

Amendments to the Claims:

In the Claims:

1. (Original) A method for detecting a defect in a computing system including one or more central processing units (CPUs) and a system memory configurable to de-allocate defective portions thereof, the method comprising:

loading a test program code into an area of the system memory to be tested, the test program code having a plurality of instructions configured to detect one or more defects in the system memory;

fetching an instruction of the test program code from the system memory;

executing the fetched instruction with a CPU; and

determining whether the executed instruction yields a test result in conformance with an expected result.

2. (Original) The method of claim 1, further comprising reporting the test result.

3. (Original) The method of claim 1, further comprising:

de-allocating a portion of the system memory upon determining that the portion is defective; and

removing a CPU upon determining that the CPU is defective.

4. (Original) The method of claim 1, wherein loading the test program code further comprises:

determining the area of the system memory to be test;

generating the test program code designed to detect the one or more defects, and

simulating execution of the test program code to generate the expected result.

5. (Original) The method of claim 1, wherein loading the test program code further comprises:

switching at least one CPU to protect mode;

changing a segment limit of the at least one CPU, and

switching the at least one CPU to big real mode.

6. (Original) The method of claim 1, further comprising:

before executing the fetched instruction,
installing an exception handler to branch a flow of CPU program control upon
detection of a defect;

disabling the interrupts to prevent interruption in the execution of the test program
code;

after executing the instruction,
enabling the interrupts; and
removing the exception handler.

7. (Original) The method of claim 1, wherein the expected result includes one or more
metrics.

8. (Original) The method of claim 7, wherein determining whether the executed instruction
yields a test result in conformance with the expected result further comprises comparing the test
result against at least one metric.

9. (Original) The method of claim 8, wherein the at least one metric is a numerical result.

10. (Original) The method of claim 8, wherein the at least one metric is a number of CPU
clock cycles.

11. (Original) The method of claim 8, wherein the at least one metric is a number of seconds
or a fraction thereof representing an execution run time.

12-14 (Cancelled)

15. (Original) An apparatus for use by a computer user for detecting a defect in a computing
system including one or more central processing units (CPUs) and a system memory
configurable to de-allocate defective portions thereof, wherein at least one CPU includes a code
cache and a data cache, the computing system configured to fetch an instruction of a test
program code from the system memory for testing the one or more CPU and the system memory,
the apparatus comprising:

means for detecting a first defect related to a first corrupted portion of the fetched instruction, where the first corrupt portion represents data; and

means for detecting a second defect related to a second corrupted portion of the fetched instruction, where the second corrupt portion represents code.

16. (Original) An apparatus of claim 15, further comprising:

means for removing a defective CPU from the computer system, if so detected; and

means for removing a defective portion of system memory from the computer system, if so detected.

17. (Original) An apparatus of claim 15, further comprising:

means for reporting whether the CPU and the system memory are defective to the computer user.

18. (Original) An apparatus of claim 15, wherein the first defect is due to a defective data access path.

19. (Original) An apparatus of claim 15, wherein the second defect is due to a defective code access path.

20. (Original) A computer software product for detecting a defect in a computing system including one or more central processing units (CPUs) and a system memory configurable to de-allocate defective portions thereof, the computer software product includes a medium readable by a CPU, the medium having stored thereon:

a first sequence of instructions which, when executed by the CPU, causes the CPU to load a test program code into an area of the system memory; and

a second sequence of instructions which, when executed by the CPU, causes the CPU to indicate the presence of the defect related to a code portion of a fetched instruction of the test program code; and

a third sequence of instructions which, when executed by the CPU, causes the CPU to indicate the presence of the defect related to a data portion of the fetched instruction of the test program code.

21. (Original) The computer software product of claim 20, further comprising:
a fourth sequence of instructions which, when executed by the CPU, causes at least one of the CPUs to be removed from the computer system; and
a fifth sequence of instructions which, when executed by the CPU, causes at least one portion of the system memory to be removed from the computer system.

22. (Original) A computer-readable medium having stored thereon instructions which when executed by a CPU, cause the CPU to perform steps of:

loading a test program code into an area of a system memory, the test program code having a plurality of instructions configured to detect one or more defects in the system memory;
fetching an instruction of the test program code from the system memory;
executing the fetched instruction within a CPU; and
determining whether the executed instruction yields a test result in conformance with an expected result,

wherein a nonconforming test result indicates the presence of the defect in the system memory.

23. (Canceled)

24. (Previously presented) A method for performing ongoing testing to detect defects in a computer system including one or more central processing units (CPUs) and a system memory configurable to de-allocate defective portions thereof, the method comprising:

loading a test program code into an area of the system memory to be tested, the test program code having a plurality of instructions configured to detect one or more defects in the system memory; and

periodically, fetching an instruction of the test program code from the system memory;

executing the fetched instruction with a CPU; and

responsive to the results of the execution of the test program code, determining whether instructions can be executed from the section of memory.